CENTRE DE RECHERCHE SUR LES TRANSPORTS
(C.R.T.)
CENTRE FOR RESEARCH ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE

CENTRE DE RECHERCHE SUR LES TRANSPORTS
UNIVERSITÉ DE MONTRÉAL
C.P. 6128, SUCCURSALE A
MONTRÉAL
CANADA H3C 3J7

TÉLÉPHONE: (514) 343-7575
TÉLEX: 05-24146 BIBPOLYTEC
EMAIL: crt@crt.umontreal.ca
TÉLÉCOPIEUR (FAX): (514) 343-7121

CRT-688

CHECKING UNICURSALITY OF A MIXED GRAPH: A BASE FOR AN OPTIMAL ALGORITHM FOR THE MIXED CHINESE POSTMAN PROBLEM*

by

Y. Nobert [1,2]
J.C. Picard [2]
P. Samson [2]

[1] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, Succursale A, Montréal (Québec) CANADA H3C 3J7.

[2] Département des Sciences administratives, Université du Québec à Montréal, 1495, rue Saint-Denis, Montréal (Québec) CANADA H3C 3P8.

* This work was supported by the Canadian Natural Sciences and Engineering Research Council (Grants OGP0005486 and OGP0005811).

Centre de recherche sur les transports - Publication #688

April 1990

90-04328/

Dépôt légal - 2ième trimestre 1990

ABSTRACT

The Chinese postman problem is well solved when the original graph contains only arcs or only edges. The mixed Chinese postman problem is however NP-complete and very few papers are devoted to this case. In this paper we present an efficient way to check the unicursality of a mixed graph which enables us to calculate a sharp lower bound on the optimal value of the mixed Chinese postman problem using the simplex method. An algorithm is also proposed to achieve optimality by using Gomory cuts and blossom inequalities.

Key words :Mixed postman unicursality, simplex.

- - - - - - -

RÉSUMÉ

Le problème du postier chinois peut être résolu par un algorithme polynômiale lorsque le graphe original est complètement orienté ou complètement non-orienté. Le problème du postier chinois mixte (MCPP) est toutefois NP-complet et on retrouve dans la littérature peu d'articles le concernant. Le présent article décrit un algorithme nouveau et efficace pour vérifier si un graphe mixte est Eulerien. Cet algorithme nous permet de calculer une borne inférieure très forte sur la valeur optimale du MCPP lorsqu'il et utilisé conjointement avec la méthode du simplexe. Nous proposons aussi un algorithme optimal pour le MCPP où des coupes de Gomory et des contraintes de parité sont utilisées.

Mots-clés : Postier mixte Eulérien, optimal, simplexe.

1. INTRODUCTION

In this paper we consider the mixed Chinese postman problem abbreviated in the following by MCPP. Let $G = (N, A, E)$ be a graph where $N = \{1, \ldots, n\}$ is a set of nodes, $A = \{(i,j) : (i,j) \in NxN, i \neq j\}$ is a set of arcs and $E = \{(i,j) : (i,j) \in NxN, i < j\}$ is a set of edges. The arcs or edges of G will be termed the "links" of the graph. We suppose known non-negative lengths associated with each link of G. More precisely, c_{ij} and d_{kl} will represent respectively the lenght of arc (i,j) in A and the length of edge (k,l) in E.

The graph G is said to be unicursal or Eulerian if there exists a circuit passing through each link of G exactly once while respecting the orientation of each arc. The MCPP is the problem of adding copies of arcs or edges in G in order to render the resulting graph eulerian while minimizing the total length of the links added. If $A=\emptyset$ or $E=\emptyset$, then the MCPP reduces to the Chinese postman problem (CPP) and in both cases this problem is well solved. When G contains only arcs, solving the CPP amounts to finding a minimum cost circulation [4]. If only edges are present in G then Edmonds and Johnson [2] have shown that the CPP can be sovled by calculating a minimal weighted 1-matching between the odd nodes of G.

The MCPP however is NP-complete [8] and very few papers have been devoted to this case. Kapauf and Koehler [5] have studied the properties of a mathematical formulation of the MCPP. They do not propose any algorithm associated with their model. Minieka [6], Frederickson et al [3] as well as Edmonds and Johnson [2] have constructed approximate algorithms for the MCPP and studied for some of them their worst case. To our knowledge, no optimal algorihtm has been developed for the MCPP. This problem is however important in practice and is closely related to the problem of collecting garbage along the streets of a city [1]. The aim of this paper is to present a new way of checking unicursality of a mixed graph and to explain how it can lead to an optimal algorithm for the MCPP.

2. NECESSARY AND SUFFICIENT CONDITIONS

We suppose in this section and in the rest of the paper that G is strongly connected. We describe here two necessary and sufficient conditions for a mixed graph to be unicursal and how we can impose them in a mathematical formulation of the MCPP. Let x_{ij} and y_{ij} represent respectively an optimal number of copies of arc $(i,j) \in A$ and edge $(i,j) \in E$ which added to the graph G make it eulerian. Clearly these variables must take non-negative integer values. The graph $G_{x,y}$ denote the original graph G with the additional copies of links given by the values of x_{ij} and y_{ij}. More precisely, if $x_{ij} = p$ then $G_{x,y}$ will contain $(p+1)$ arcs of the form (i,j). A similar interpretation is given to $y_{ij} = p$. If the graph G is unicursal, then in the optimal solution of the MCPP we have $x_{ij} = y_{ij} = 0$ for all the links of G. If, however, the original graph is not Eulerian, we want to find values for the variables x_{ij} and y_{ij} such that $G_{x,y}$ is unicursal and such that the following objective is minimized:

$$Z = \sum_{(i,j) \in A} c_{ij} x_{ij} + \sum_{(i,j) \in E} d_{ij} y_{ij} \qquad (1)$$

Ford and Fulkerson [4] have shown that the two following conditions will be satisfied by $G_{x,y}$.

2.1 Evenness condition

The degree of each node in $G_{x,y}$ will be even.

In a mixed graph, the degree of a node is simply the number of links incident to it. A natural way to impose this condition is by using the following constraints:

$$\sum_{(i,j)\in A(k)} x_{ij} + \sum_{(i,j)\in E(k)} y_{ij} = 2v_k + p_k \qquad (k\in N) \quad (2)$$

$$v_k \geq 0 \text{ and integer} \qquad (k\in N) \quad (3)$$

where $p_k = 1$ if the degree of node k is odd in G (k is said odd) and $p_k = 0$ otherwise (k is said even). The sets A(k) and E(k) are respectively the sets of arcs and edges incident to node k in G.

Let S be a proper subset of N. We denote by $A(S,\bar{S})$ and $E(S,\bar{S})$ respectively the set of arcs in G and the set of edges in G with one endpoint in S and the other in $\bar{S} = N-S$. A subset S of nodes is said odd if it contains an odd number of odd nodes in the original graph G. Consider the following constraints:

$$\sum_{(i,j)\in A(S,\bar{S})} x_{ij} + \sum_{(i,j)\in E(S,\bar{S})} y_{ij} \geq 1 \quad \forall S \subset N \text{ and } S \text{ odd} \quad (4)$$

The constraints (4) constitute a direct generalization to a mixed graph of the so-called "blossom inequalities" derived by Edmonds and Johnson [2] in the case of a non-oriented graph.

It is easily seen that S is odd if and only $|A(S,\bar{S})| + |E(S,\bar{S})|$ is also odd. Since in any Euler tour the frontier between S and $\bar{S}$ will be crossed an even number of times, the blossom inequalities are valid for the MCPP. Even though the constraints of (4) are quite strong, they are not sufficient in the context of the MCPP to ensure the eveness condition as it is the case for the non-oriented CPP.

The constraints (2) and (3) are in polynomial number in n whereas constraints (4) are in exponential number in n. Nevertheless we choose in the algorithm described later to impose mainly a subset of the constraints defined by (4). If the need arises, we may have to generate some constraints of (2) and (3). For reasons explained later, this is more efficient and very few additional variables such as the v_i's in (2) have to be defined.

Consider the graph depicted in figure 1.

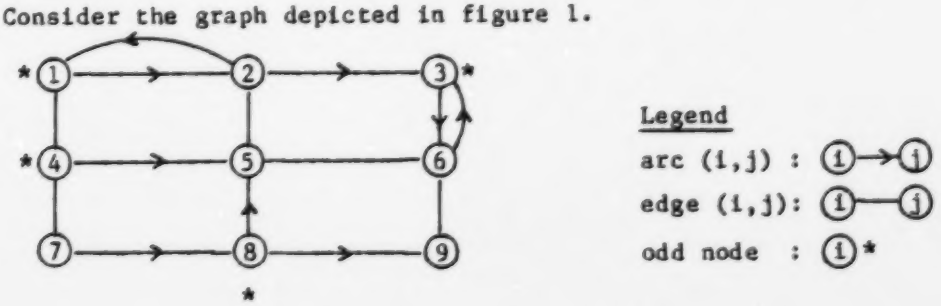


Figure 1: A mixed graph and the blossom inequalities

The following constraints constitute some blossom inequalities that we can derive for this graph:

$$x_{12} + x_{21} + y_{14} \geq 1 \qquad (B_1)$$
$$x_{45} + y_{14} + y_{47} \geq 1 \qquad (B_4)$$
$$x_{12} + x_{21} + x_{45} + x_{78} + x_{85} + x_{89} + y_{47} \geq 1 \qquad B(1, 4, 8)$$

In general, we denote by B(S) the blossom inequality associated with the odd set $S \subset N$. When $S = \{i\}$, we simply write B_i.

The constraint B(S) is "violated" whenever N contains an odd subset in the sense that $x_{ij} = 0$ for all arcs and $y_{ij} = 0$ for all edges (called the null solution) is not feasible for this constraint.

2.2 Balanced sets condition

In $G_{x,y}$, for all proper subsets S of N, the number of arcs going out of S minus the number of arcs going in S will be less than or equal to the number of edges linking S and $\bar{S}$.

In order to write this condition in a mathematical form, we need the following definitions:

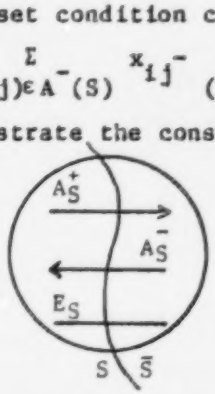
$$A^+(S) = \{(i,j) \in A : i \in S \text{ and } j \in \bar{S}\}.$$
$$A^-(S) = \{(i,j) \in A : j \in S \text{ and } i \in \bar{S}\}.$$

We will also write $D_S^+ = |A^+(S)|$, $D_S^- = |A^-(S)|$ and $D_S = |E(S)|$.

The balanced set condition can now be imposed by:

$$\sum_{(i,j) \in A^+(S)} x_{ij} - \sum_{(i,j) \in A^-(S)} x_{ij} - \sum_{(i,j) \in E(S)} y_{ij} \leq D_S^- + D_S - D_S^+, \quad \forall S \subset N \quad (5)$$

In Figure 2, we illustrate the constraints of (5).



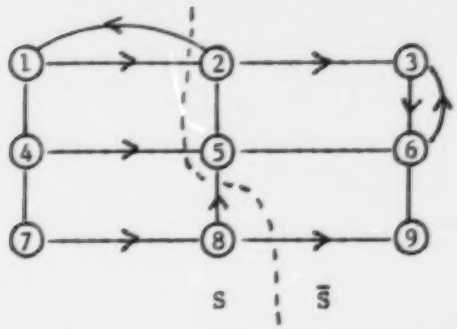
Legend:

$A^+(S)$: S ⟶ $\bar{S}$
$A^-(S)$: S ⟵ $\bar{S}$
$E(S)$: S ⟷ $\bar{S}$

Figure 2: Balanced sets condition and constraints (5)

Let $k_S = D_S^- + D_S - D_S^+$. If $k_S < 0$ then we say that S is unbalanced by $u_S = -k_S$ and at least u_S links must be added in G from $\bar{S}$ to S (u_S copies of arcs in $A^-(S)$ or edges in E(S)). If $k_S > 0$, S is not unbalanced but, if no copies of links from $\bar{S}$ to S are present in $G_{x,y}$, at most k_S arcs in $A^+(S)$ can be added in order to keep the set S balanced. When $E(S) = \emptyset$ then (5) can be written as an equation. It is easily seen by considering in this case the particular constraints of (5) associated with S and S'=N-S. The constraints of (5) will be denoted hereafter by uB(S) for $S \subset N$ and by uB_i when $S = \{i\}$. It is important to note here that a set S with $(D_S^- - D_S - D_S^+) < 0$ is also unbalanced but do not violate (5). It is because (5) only deal with outward unbalanced sets. In order that the balanced sets condition is satisfied it is however sufficient to consider only the constraints of (5). It is by no means obvious how to find an unbalanced set in a mixed graph. We will describe in the next section a very elegant way to detect all the most unbalanced sets in G, that is the sets $S \subset N$ such that u_S is maximal.

We illustrate in figure 3 the concept of unbalanced sets.



- $S=\{1,4,7,8\}$ is unbalanced and $u_S=3$
- node 8 is unbalanced and $u_8=1$

Figure 3: Unbalanced sets

We will denote by u_i the value u_S when $S=\{i\}$. In figure 3, we see that $u_8 = +1$. This means that the node 8 is unbalanced by 1. It is easily understood that at least one copy of arc (7,8) must be added to G ($x_{78} \geq 1$). We also remark in figure 3 that the set $S=\{1,4,7,8\}$ is unbalanced by 3. Since in this particular graph $E(S)=\emptyset$ and $A^-(S)=\{(2,1)\}$ then it is true that $x_{21} \geq 3$. The set $\{1,4,7,8\}$ is a most unbalanced set in this graph (as shown later). The set $\{1,2,4,7,8\}$ is also unbalanced by 3 and thus constitutes also a most unbalanced set. The following constraints are valid for the MCPP associated with the graph depicted in figure 3:

$$x_{36} \geq 1,\ x_{21} \geq 3,\ y_{25} \geq 3\ .$$

If we look more closely at the figure 3, we see that the node 3 is also unbalanced but $k_3=1$. This is because node 3 is unbalanced inward. Clearly however we have $u_{N-\{3\}} = +1$. It is thus true that $x_{36} \geq 1$ in the optimal solution. In the algorithm that follows we will impose a constraint uB_i for each outward unbalanced node and a constraint $uB_{N-\{i\}}$ for each inward unbalanced node.

Again it should be noted here that the null solution is not feasible whenever a graph contains unbalanced sets. According to the previous explanations, a valid mathematical formulation of the MCPP is:

$$P \left\{ \begin{array}{ll}
\text{Min } Z = \sum_{(i,j)\in A} c_{ij} x_{ij} + \sum_{(i,j)\in E} d_{ij} y_{ij} & (6) \\
\text{subject to:} & \\
\sum_{(i,j)\in A(k)} x_{ij} + \sum_{(i,j)\in E(k)} y_{ij} = 2v_k + p_k \qquad (k\in N) & (7) \\
\sum_{(i,j)\in A(S)} x_{ij} + \sum_{(i,j)\in E(S)} y_{ij} \geq 1 \qquad (S \subset N,\ S \text{ odd}) & (8) \\
\sum_{(i,j)\in A^+(S)} x_{ij} - \sum_{(i,j)\in A^-(S)} x_{ij} - \sum_{(i,j)\in E(S)} y_{ij} \leq k_S \quad (S \subset N) & (9) \\
x_{ij} \geq 0 \text{ and integer} \qquad ((i,j)\in A) & (10) \\
y_{ij} \geq 0 \text{ and integer} \qquad ((i,j)\in E) & (11) \\
v_i \geq 0 \text{ and integer} \qquad (i\in N) & (12)
\end{array} \right.$$

The blossom inequalities (8) are redundant relatively to (7), (10), (11), and (12). We will see however that (7) is relaxed initially in our algorithm and thus (8) are very useful in order to impose the evenness condition.

If the original graph is unicursal, $p_k=0$ for all nodes k, there are no odd sets and $k_S>0$ for all subsets S of N. In this case, $v_k=0$ for all k and the null solution for x and y is feasible for P. Since we suppose $c_{ij}>0$ and $d_{ij}>0$ for all links, then this solution is also optimal.

The model P contains an exponential number of constraints. We will see in the next section that P can however be tackled with very efficiently if strong violated constraints can be identified in an initial relaxation of P. Other models with a polynomial number of constraints have been studied in the literature. They have not led to any efficient exact algorithm.

3. AN OPTIMAL ALGORITHM FOR THE MCPP

3.1 The algorithm is general

Due to the large number of constraints in P, it is impossible even for small n to solve this problem directly by writing down explicitly all the constraints and using a branch-and-bound or a cutting planes code. Instead we will recourse to the following algorithm. Before describing it in all details, we will introduce some supplementary notation:

P_k is the relaxation of P at iteration k of the algorithm;

$v(Q)$ denotes the optimal value of problem Q;

m_k is the number of constraints in problem P_k;

x^k, y^k constitute one optimal solution of problem P_k.

The algorithm goes as follows:

Step 0: {**Initial phase**}

- Identify in G all the odd nodes, all the unbalanced nodes (inward or outward) and all the most unbalanced sets.
- Construct P_o as the linear program with the objective function (6), the blossom inequalities associated with odd nodes and the constraints of (9) corresponding to the unbalanced nodes and sets identified previously.
- Set m_o equal to the number of constraints generated and set k=0.

Step 1: {**Simplex phase**}

- Evaluate x^k, y^k, and $v(P_k)$ by solving P_k with the simplex method.

Step 2: {**Identification of other violated constraints**}

- Identify violated constraints of (8) and (9) by analyzing G_{x^k,y^k}.
- Let p be the number of constraints found.
- If p=0 then go to step 3.
  Else . add explicitly the p new constraints to P_k
  . set k=k+1 and $m_{k+1}=m_k+p$
  . go to step 1.

Step 3: {**Adding Gomory cuts**}

- If (7), (10), (11), and (12) are satisfied, then go to step 4.
- If some constraints of (10) or (11) are violated, add one Gomory cut and go to step 1.
- Identify an odd node in G_{x^k,y^k}.
- Generate the appropriate constraint of (7) and the corresponding v_i variable.
- Add a Gomory cut on v_i to the model and go to step 1.

Step 4: {**Identification of the optimal solution**}

- Set $v(P)=v(P_k)$, $x=x^k$ and $y=y^k$.
- Determine the Euler tour in G_{x^k,y^k}. (One can use the algorithm described in [2]).
- END.

The previous algorithm is a dual method where the first feasible solution is optimal. Its efficiency mainly depends on two factors. Firstly, we must be able to solve rapidly the different relaxations encountered in step 1. If the total number of constraints generated stays reasonably small, the simplex method will perform in an efficient way. Secondly, the effective constraints defining the optimal extreme point must be identified early in the process. If this should not happen, the number of constraints generated will grow large and hence a lot of time will be spent in simplex iterations. Furthermore, numerical instability could set in. The best guarantee against this bad situation is to always generate strongly violated constraints. This should ensure that $v(P_k)$ will converge to $v(P)$ rapidly. Usually this kind of method works very well or does not work at all. In [7], Nobert has studied many variants of a similar approach to solve exactly extensions of the symmetrical traveling salesman problem. The efficiency of this relaxation process has been proven by many authors on different combinatorial optimization problems. We now revert to more specific comments concerning the algorithm described in this section.

3.2 Specific comments

In order that the algorithm given in section 3.1 can be implemented, we must be able to circumvent two difficulties:

1. Identify violated blossom inequalities.
2. Identify most unbalanced sets.

These two difficulties are not trivial at all to resolve. The way we achieve this can be considered as one of the main scientific results of this paper.

The third difficulty is the easiest one. Let us suppose that we are at iteration k of the algorithm and that G_{x^k,y^k} is analyzed. Denote by $D^+_{S(x^k,y^k)}$, $D^-_{S(x^k,y^k)}$, $D_{S(x^k,y^k)}$ and $k_{S(x^k,y^k)}$, the new values of respectively D^+_S, D^-_S, D_S, and k_S in G_{x^k,y^k}. Checking parity amounts to

counting the number of links incident to each node. If every node of G_{x^k,y^k} are even and $k_{S(x^k,y^k)} > 0$ for all $S \subset N$, then G_{x^k,y^k} is Eulerian and represents the optimal solution of the MCPP. A subtle point is in order here. When an unbalanced set S is found in G_{x^k,y^k}, we generate a constraint of (9) whose right-hand side is k_S and not $k_{S(x^k,y^k)}$. This constraint is still violated, cuts the actual solution and, more importantly, is valid for the MCPP. Using $k_{S(x^k,y^k)}$ as the right-hand side would render the proposed algorithm in 3.1 sub-optimal. We now explain how to identify violated blossom inequalities and most unbalanced sets in a graph.

Identifying violated blossom inequalities

Let $G^A_{x^k,y^k} = (N, E_{x^k,y^k})$ be a non-oriented graph where $E_{x^k,y^k} = \{(i,j) : i<j, (x^k_{ij}>0 \text{ or } x^k_{ji}>0 \text{ or } y^k_{ij}>0)\}$. There is an edge (i,j) in E_{x^k,y^k} if (i,j) or (j,i) is a link in G and if the corresponding variable x^k_{ij}, x^k_{ji}, or y^k_{ij} takes a positive (possibly fractional) value in the optimal solution of P_k.

Let $S_1, \ldots, S_a$ be the connected components of $G^A_{x^k,y^k}$. Clearly, for any ℓ such that S_ℓ is odd the constraint B_{S_ℓ} is violated by x^k, y^k and can be appended to our relaxation in order to cut off the solution x^k, y^k. If, however, every set S_ℓ is even, then no violated blossom inequalities can be derived directly from these sets. Calculating the connected components requires $O(|A| + |E|)$ operations and this procedure is thus quite efficient.

Identifying most unbalanced sets

Let $\tilde{G}=(\tilde{N}, \tilde{A}, \tilde{E})$ be an arbitrary mixed graph where $\tilde{a}_{ij}$ and $\tilde{u}_{k\ell}$ respectively represent the number of arcs of the form (i,j) in $\tilde{A}$ and the number of edges of the form (k,ℓ) in $\tilde{E}$. To this graph $\tilde{G}$ we associate a quadratic function $f_{\tilde{G}}$ defined as follows:

$$f_{\tilde{G}}(x_1, \ldots, x_n) = \sum_{(i,j)\in\tilde{A}} \tilde{a}_{ij}\, x_j(1-x_i) + \sum_{(i,j)\in E} \tilde{u}_{ij}\,(x_i-x_j)^2 - \sum_{(i,j)\in\tilde{A}} \tilde{a}_{ij}\, x_i(1-x_j) . \qquad (13)$$

We will denote by $f_{\tilde{G}}(S)$ the value taken by $f_{\tilde{G}}(x_1, \ldots, x_n)$ when, for $S \subset N$, we let $x_i=1$ if $i \in S$ and $x_i=0$ otherwise.

It is easily seen that:

$$f_{\tilde{G}}(S) = D_S^- + D_S - D_S^+ = k_S \qquad (14)$$

with D_S^+, D_S^-, D_S, and k_S defined as earlier but for the graph $\tilde{G}$. $f_{\tilde{G}}$ will take a negative value only when the set S is unbalanced outwardly in $\tilde{G}$. If $S=\emptyset$ and so $x_i=0$ for all i, we have $f_{\tilde{G}}(S)=0$ hence (15) is true:

$$f^*_{\tilde{G}} = \min_{S \subset \tilde{N}} f_{\tilde{G}}(S) \leq 0 . \qquad (15)$$

We conclude from this that $\tilde{G}$ does not contain any unbalanced set if and only if $f^*_{\tilde{G}} = 0$.

The minimum value of $f_{\tilde{G}}(S)$ over $S \subset N$ can be found following the work of Picard and Ratliff [9]. They showed that it is equivalent to solving a maximum flow problem on an associated graph. They also developed a procedure that enables one to determine all the sets S such that $f_{\tilde{G}}(S) = f^*_{\tilde{G}}$. Since the maximum flow problem is well-solved, it is relatively easy to derive, using this idea, all the most unbalanced sets in an arbitrary graph [10].

As an example, the graph illustrated in figure 1 has the following quadratic function associated with it:

$$f(S) = x_1 + x_3 + x_4 + 4x_5 + 2x_6 - x_8 + 2x_9 \qquad (16)$$
$$- 2x_1x_4 - 2x_4x_7 - 2x_2x_5 - 2x_5x_6 - 2x_6x_9 .$$

The function in (16) with x_i binary for all i is minimized when $x_1 = x_4 = x_7 = x_8 = 1$ and $x_2 = x_3 = x_5 = x_6 = x_9 = 0$. So the set $S_1 = \{1,4,7,8\}$ is a most unbalanced set with $u_{S_1} = -f(S_1) = 3$. For $S_2 = \{1,2,4,7,8\}$ we have also $f(S_2) = -3$ and hence S_2 is another most unbalanced set. Moreover, we can prove that S_1 and S_2 are the only most unbalanced set in this example. We refer the reader to [9] for precise details showing how all the global minimum of $f_{\tilde{G}}(S)$ can be found.

3.3 Illustration of the algorithm

Figure 4 shows the graph on which the algorithm of section 3.1 will be applied.

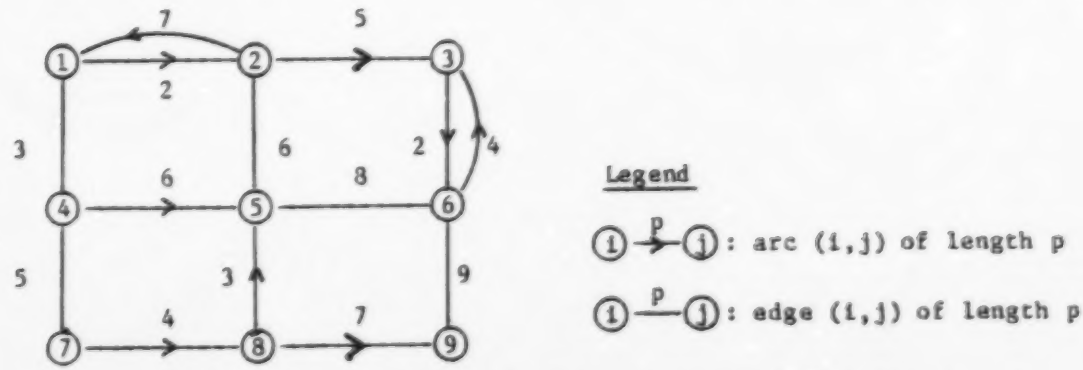


Figure 4: Graph on which the algorithm is applied

At each iteration of the algorithm a figure will illustrate the graph $G^{A}_{x^k, y^k}$ or the graph G_{x^k, y^k} with the appropriate information corresponding to the violated constraints added explicitly to the model.

{ Initial phase}

Figure 5 shows the violated constraints identified in the initial phase.

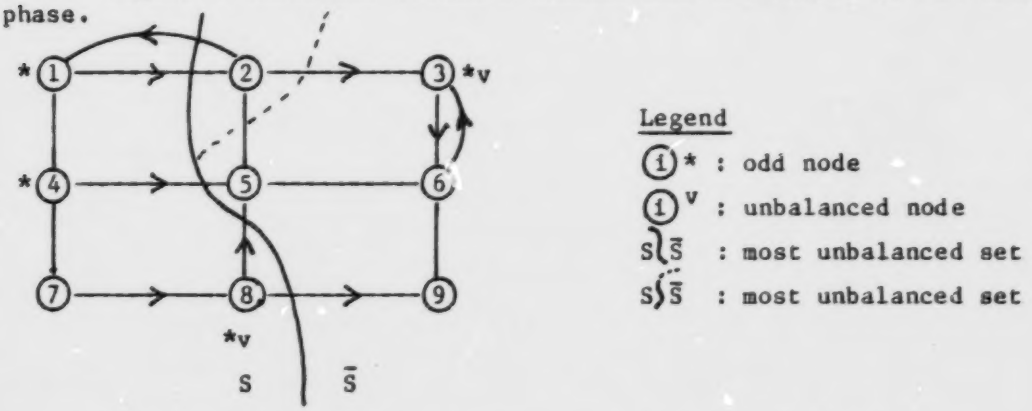


Figure 5: Violated constraints derived in the initial phase

We see in figure 5 that nodes 1, 3, 4, and 8 are odd, that node 3 is unbalanced inward and that node 8 is unbalanced outward with $u_8 = u_{N-\{3\}} = 1$. Also shown are the two most unbalanced sets in this graph, namely the sets $S_1=\{1,4,7,8\}$ and $S_2=\{1,2,4,7,8\}$ with $u(1,4,7,8) = u(1,2,4,7,8) = 3$. The initial model solved by the simplex method contains the following six constraints:

$$\boxed{uB_{N-\{3\}},\ uB_8,\ uB(1,4,7,8),\ uB(1,2,4,7,8),\ B_1,\ B_4} \qquad (17)$$

Note that B_3 and B_8 are also violated but they are redundant to $uB_{N-\{3\}}$ and uB_8.

{ Simplex phase}

The values of x^0 and y^0 are indicated in figure 6. The objective function has a value $v(P_0) = 48$ which is a lower bound on the optimal value of the MCPP.

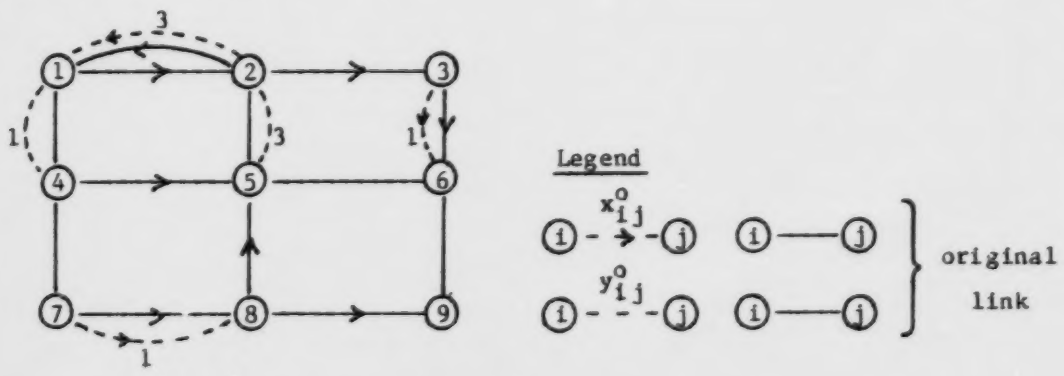


Figure 6: Graph G_{x^0, y^0}

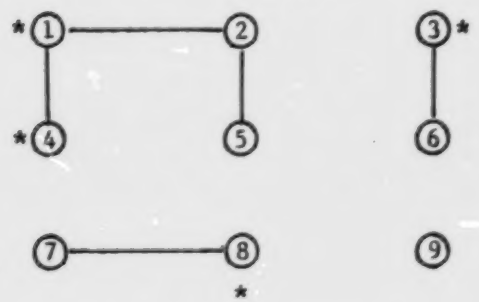
The optimal solution is thus:

$$x_{21} = y_{25} = 3,\ y_{14} = x_{78} = x_{36} = 1.$$

{Other violated constraints}

It is easily seen from figure 6 that nodes 1 and 7 are unbalanced in G_{x^0,y^0} and so the constraints uB_1, and uB_7 are violated by this solution. By minimizing $f_{G_{x^0,y^0}}(S)$ we find that $f_{G_{x^0,y^0}} = -2$ for $S=\{2,5,7\}$ and so $uB(2,5,7)$ is added to the problem.

The graph $G^A_{x^0,y^0}$ is depicted in figure 7.



Legend

① * : odd node

① — ① : $x^0_{ij} > 0$ or $x^0_{ji} > 0$ or $y^0_{ij} > 0$.

Figure 7: Graph $G^A_{x^0,y^0}$

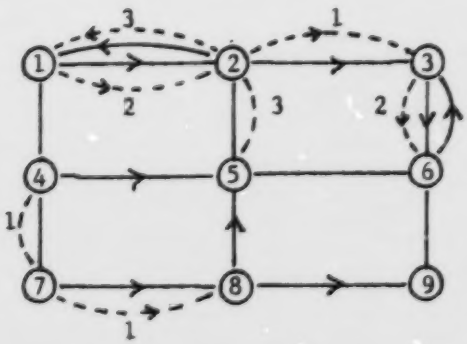
Graph $G^A_{x^0,y^0}$ has four connected components: $S_1=\{1,2,4,5\}$, $S_2=\{3,6\}$, $S_3=\{7,8\}$ and $S_4=\{9\}$. The sets S_2 and S_3 are odd in G so the constraints B(3,6) and B(7,8) are violated.

In conclusion, the following five constraints are added to the model

$$\boxed{uB_1,\ uB_7,\ uB(2,5,7),\ B(3,6),\ B(7,8)} \qquad (18)$$

{Simplex phase}

The problem now contains 11 constraints. Using the simplex method, we obtained the results shown in figure 8 with $v(P_1) = 61$.



Legend

As in figure 6.

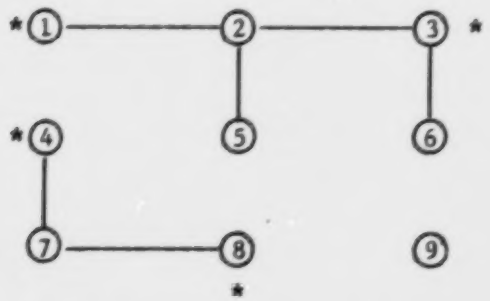
Figure 8: Graph G_{x^1,y^1}

The optimal solution of the relaxed problem is then:

$$x_{21} = y_{25} = 3,\ x_{12} = x_{36} = 2,\ x_{23} = y_{47} = x_{78} = 1.$$

{**Other violated constraints**}

Graph G_{x^1,y^1} has $f^*_{G_{x^1,y^1}} = -2$ for the set $S=\{4,7\}$ and so the constraint uB(4,7) is violated. This constraint is thus added to the model. Figure 9 shows the connected components of $G^A_{x^1,y^1}$:



Legend

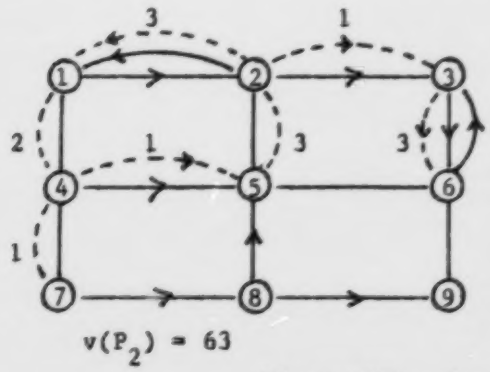
As in figure 7.

None of the connected components of $G^A_{x^1,y^1}$ are odd and no violated blossom inequalities are identified. Note that the node 2 in G_{x^1,y^1} is however odd. This illustrates the fact that the blossom inequalities are not sufficient alone to ensure the evenness condition. In a situation like this with no unbalanced sets, a v_i variable and the appropriate constraint of (7) would have to be appended to the model. A Gomory cut derived from this variable would then cut the infeasible solution. In résumé, only the following constraint is added to the model at this stage of the algorithm:

$$\boxed{uB(4,7)} \qquad (19)$$

{**Simplex phase**}

After reoptimizing in order to take account of the constraint in (19), the simplex method proposes the following solution:



Legend

As in figure 8.

Figure 10: Graph G_{x^2,y^2}

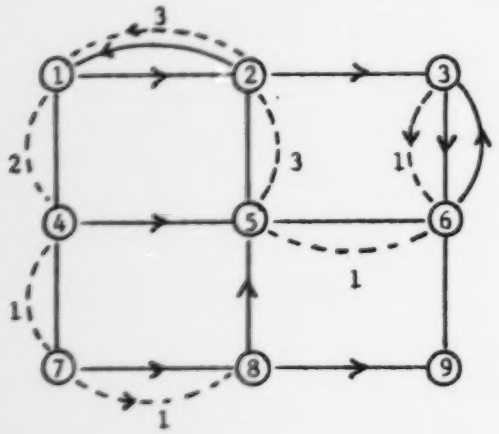
{Other violated constraints}

As previously, no violated blossom inequality can be identified in $G^A_{x^2,y^2}$. However, the node 2 is unbalanced in G_{x^2,y^2} and the set $S=\{2,4,7\}$ is a most unbalanced set in this graph. We then add explicitly the constraints of (20) in the model:

$$\boxed{uB_2,\ uB(2,4,7)} \qquad (20)$$

{Simplex phase}

There are now 14 constraints involved in the linear program. Its optimal solution has a value of $v(P_3) = 64$ and implies graph G_{x^3,y^3} depicted below.



Legend

As in figure 6.

Figure 11: Graph G_{x^3,y^3}

One can easily verify that every node in G_{x^3,y^3} is even and that there are no odd connected components in $G^A_{x^3,y^3}$. Moreover, $f^*_{G_{x^3,y^3}} = 0$ so we can conclude that G_{x^3,y^3} is Eulerian. Since every constraint of P_3 is valid for the MCPP associated with G, we can claim also that G_{x^3,y^3} gives an optimal way of adding copies of links in G in order to render it Eulerian.

It is important to notice that the mathematical model P (page 4) associated with the graph depicted by figure 4 contains 9 v_i variables, 14 x_{ij} and y_{ij} variables and a number of constraints proportional to $2^9=512$. We have found the optimal solution of this (quite difficult) integer linear program using only 14 constraints, no v_i variables, no Gomory cuts and a very small number of simplex iterations. For those who are curious, one Euler tour in G_{x^3,y^3} is (1-2-3-6-3-6-5-2-1-4-7-8-9-6-5-2-1-4-7-8-5-2-1-4-5-2-1). Initially, 14 links were present in G and 12 more links were needed to make it Eulerian at minimum length.

4. CONCLUSION

In this paper, we have presented a new way of checking unicursality of a mixed graph. This enables us to propose a new model and a new algorithm for the MCPP. This algorithm, beginning with a linear relaxation of the problem, successively adds new violated constraints until a feasible and hence optimal solution is obtained. The originality of the proposed method lies in the way unbalanced sets and violated blossomm inequalities are identified. The simplex method and a max flow algorithm are the two principal tools we need to implement the algorithm. We are now performing exhaustive computer tests to evaluate the global efficiency of the method.

Right now we can affirm that the results are quite promising. The main advantage of this method lies in its flexibility. Virtually any pertinent linear constraint as well as lower or upper bounds on the variables can easily be tackled with. We envisage in the future to use this method to solve many extensions of the MCPP.

REFERENCES

[1] Bodin, L.D., Kursh, S.J., "A Computer-Assisted System for the Routing and Scheduling of Street Sweepers", Operations Research 26, 525-537, 1978.

[2] Edmonds, J., Johnson, E.L., "Matching, Euler Tours and the Chinese Postman", Mathematical Programming 5, 88-124, 1973.

[3] Frederickson, G.N., Hecht, M.S., Kim, C.E., "Approximation Algorithms for Some Routing Problems", SIAM Journal of Computing 7, 178-193, 1978.

[4] Ford, L.R., Fulkerson, D.R., "Flows in Networks", Princeton University Press, 1962.

[5] Kapauf, C.H., Koehler, G.J., "The Mixed Postman Problem", Discrete Applied Mathematics 1, 89-103, 1979.

[6] Minieka, E., "The Chinese Postman Problem for Mixed Networks", Management Science 25, 643-648, 1979.

[7] Nobert, Y., "Construction d'algorithmes optimaux pour des extensions au problème du voyageur de commerce", Ph.D. thesis, Université de Montréal, 1983.

[8] Papadimitriou, C.H., "On the Complexity of Edge Traversing", Journal of the Association for Computing Machinery 23, 544-554, 1976.

[9] Picard, J.-C., Ratliff, H.D., "Minimum Cut and Related Problems", Networks 5, 4, 1975.

[10] Picard, J.-C., Queyranne, M., "On the Structure of all Minimum Cuts in a Network and Applications", Mathematical Programming 13, 1980.